PortShare

User Manual

Rev: 1.3

August24th 2010

Opengear's serial port sharing software (*PortShare*) delivers the virtual serial port technology your Windows and Linux applications need to open remote serial ports and read the data from serial devices that are connected to your *console server*.

PortShare is supplied free with each *console server* and you are licensed to install *PortShare* on one or more computers for accessing any serial device connected to any Opengear *console server* port. So one user can access multiple remote serial devices in multiple locations, and multiple users can access the same remote serial device.

Publishing history

Date	Revision	Update details
Oct 2009	1.0	Initial release of PortShare (Windows)
Dec 2009	1.1	Linux driver documentation added
Feb 2010	1.2	License agreements included
Aug 2010	1.3	Encrypted Windows added

Copyright

©Opengear Inc. 2010. All Rights Reserved.

Information in this document is subject to change without notice and does not represent a commitment on the part of Opengear. Opengear provides this document “as is”, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose.

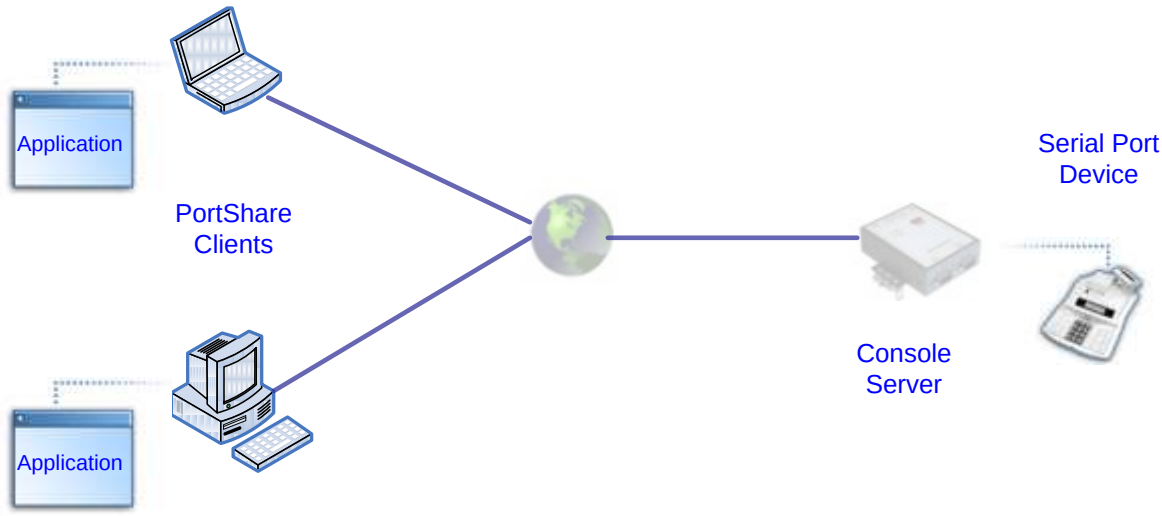
Opengear may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time. This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

1. *PortShare* for Windows

PortShare establishes secure client-server connections between the serial ports on remote *console servers* and applications on your Windows PC or server servicing COM ports.

Once connection is established, all data sent to the nominated COM port on the Windows computer will be immediately redirected and delivered out the corresponding serial port on the *console server*. Similarly incoming data on the *console server* serial port is redirected to the virtual COM port on the Windows computer where it can be processed further.

You are licensed to install *PortShare* on one or more computers for accessing any serial device connected to any Opengear *console server* port.



1.1 Install the *PortShare* client

PortShare is fully compatible with 32 bit and 64 bit versions of Windows NT 4.x, Windows XP, Windows 2000, Windows 2003, Windows 2008, Windows Vista and Windows 7. The license agreement is included in Chapter 3.

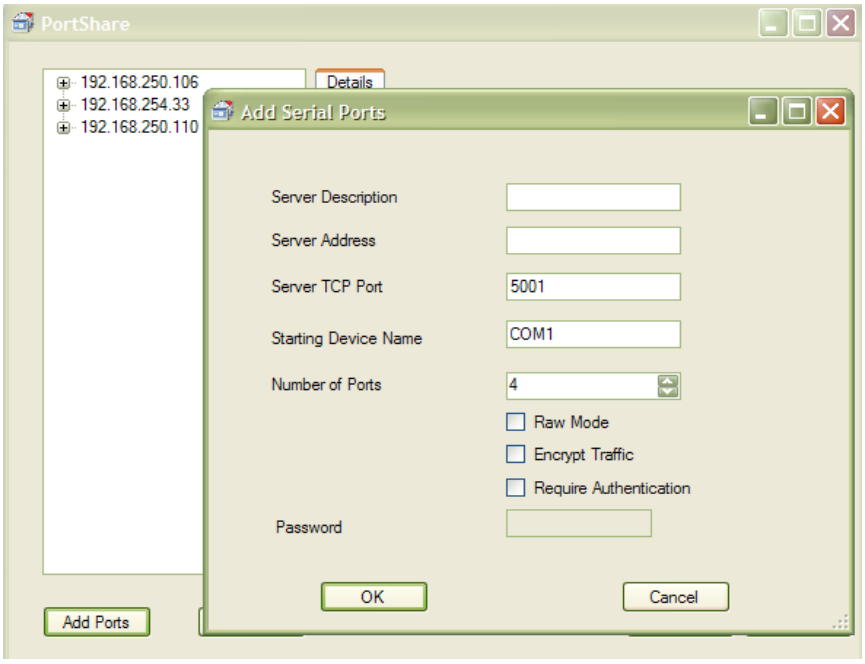
- The *portshare_setup.exe* program is included on the CD supplied with your *console server* (or a copy can be freely downloaded from ftp://ftp.opengear.com.) Double click the *portshare_setup.exe* file to start the installation process

- Read the License Agreement, then follow the prompts to select the destination path and choose the shortcuts you wish to create. Once the installer completes you will have a working *PortShare* client installed on your machine and an icon on your desktop
- Click the *PortShare* icon on your desktop to start the client

1.2 Configure the *PortShare* client

Creating the *PortShare* client connection will initiate a virtual serial port data redirection to the remote *console server* using TCP/IP protocol

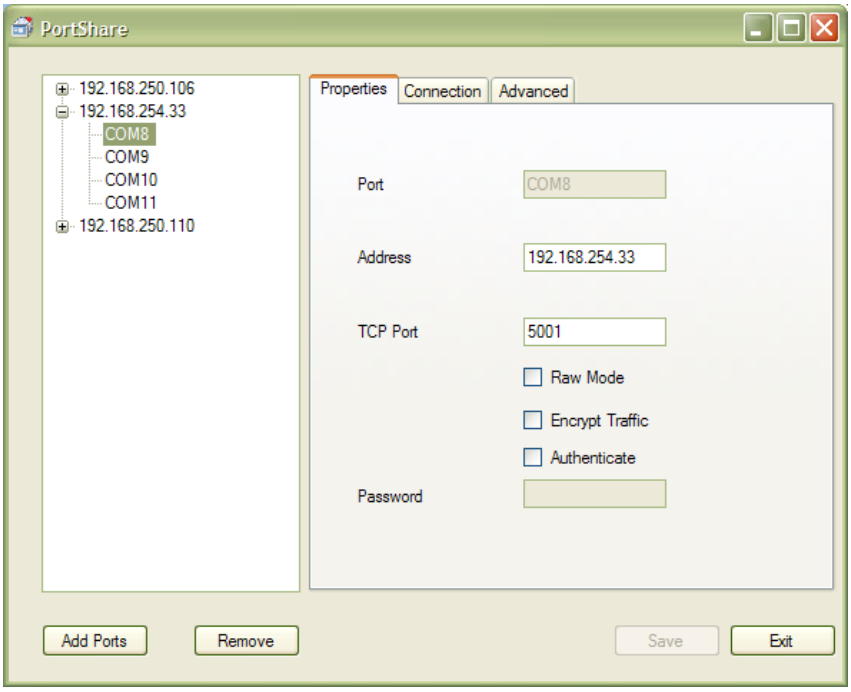
- Click on *Add Ports*
- Specify a name to identify this connection in the *Server Description* field



- Enter the *console server's* IP address (or network name) in the *Server Address* field
- Enter the *Server TCP Port* number that matches the port you have configured for the serial device on the remote *console server*. Ensure this port isn't blocked by a firewall
    - Telnet *RFC2217* mode is configured by default so the range of port numbers available on a 16 port *console server* would be 5001-5016
    - Alternately, check *RAW Mode* to set *Server TCP Port to* 4001- 4048 on a 48 port *console server*
- Select the starting COM port for the *Starting Device Name*. This can be in the range of COM1 to COM4096
- Specify the number of COM ports you want to add in *Number of Ports*. Sequential port numbers will be assigned automatically however, if a COM port number is already being used by other applications than that port number will be skipped
- Check the *Encrypt Traffic* box to enable SSL/TLS encryption of the data going to the port
- To enable authentication, check the *Require Authentication* box. You will be prompted to enter a Password
- Click OK to add the specified COM ports

1.2.1 Configure the COM *Properties*

- To configure a COM port you have created, simply click on the desired COMx label in the left hand menu tree. For example, click on COM8 from the 192.168.254.33 server. This will automatically display the Properties tab on the right-hand side of the *PortShare* window



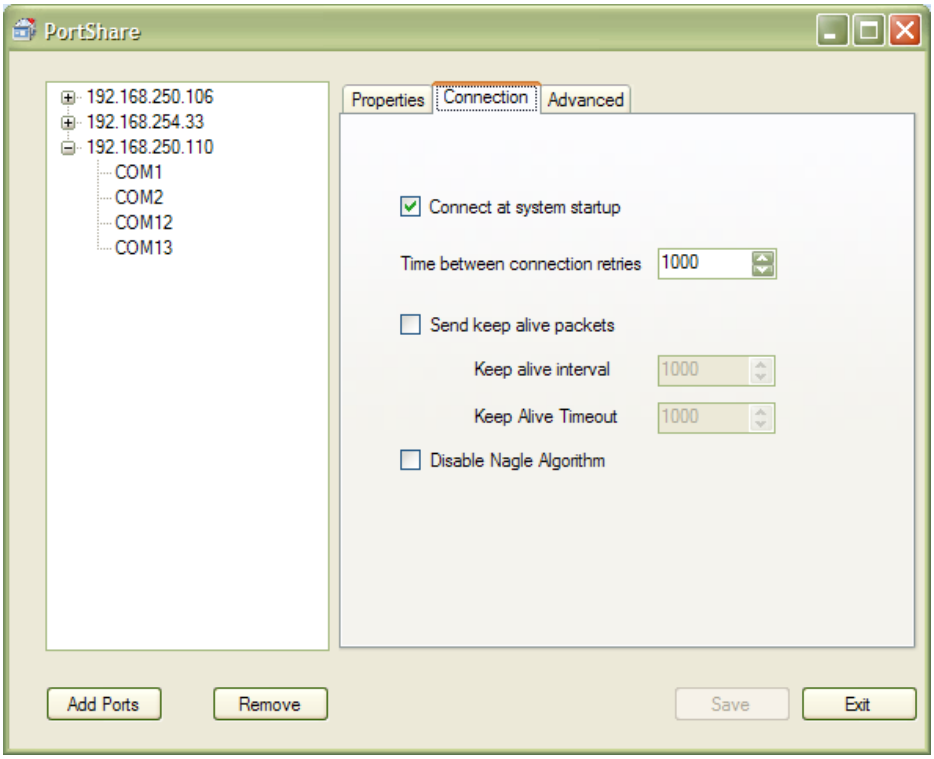
- In the Properties window, you can edit the *IP Address* or *TCP Port* to be used to connect to that COM port
- Check *Raw Mode* to change the *TCP Port* to 4001 in the above example
- Check the *Encrypt Traffic* box, to encrypt traffic on this COM port only.
- Check the *Authenticate* box, to enable authentication on this COM port. This will prompt you to enter a *Password*.
- You can also configure the COM port in the *Connection* and *Advanced tabs*

1.2.2 Configure the COM *Connection*

- Check *Connect at system startup* to connect to the *console server* when the *PortShare* service starts as opposed to waiting for the application to open the serial port before initiating the connection to the *console server*

- The *Time between connection retries* specifies the number of milliseconds between TCP connection retries after a client-initiated connection failure. Valid values are 1000-10000. The default is 1000 milliseconds or 1 second, and *PortShare* will continue attempting to reconnect forever to the *console server* at this interval

- The *Send keep alive packets* option tests if the TCP connection is still up when no data has been sent for a while by sending keep-alive messages. Select this option and specify the period of time (in milliseconds) after which *PortShare* sends a command to the remote *console server* end in order to verify the connection's integrity and keep the connection alive. This option is not available when *Raw Mode* is enabled.
    - The *Keep alive interval* specifies the number of milliseconds to wait on an idle connection before sending a keep-alive message. The default is 1000 milliseconds or 1 second.
    - The *Keep Alive Time*out specifies how long *PortShare* should wait for a keep alive response before timing out the connection
- *Disable Nagle Algorithm* — the Nagle Algorithm is enabled by default and it reduces the number of small packets sent by *PortShare* across the network

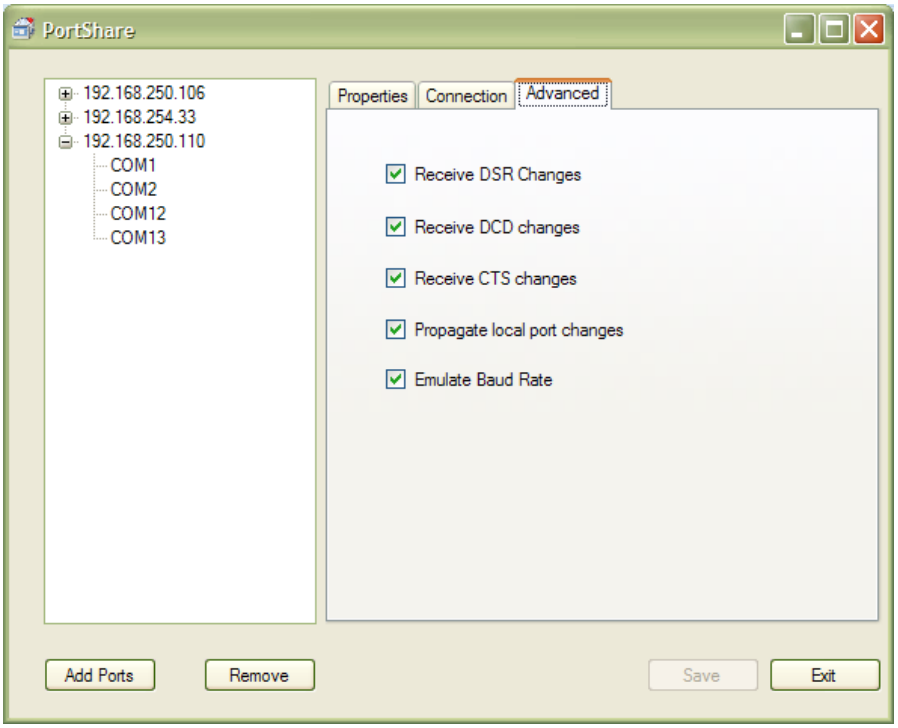


1.2.3 Configure COM *Advanced* Settings

- Check *Receive DSR/DCD/CTS changes* if the flow control signal status from the physical serial port on the *console server* is to be reflected back to the Windows COM port driver. Ssome serial communications applications prefer to run without any hardware flow control i.e. in “two wire” mode

- The *Propagate local port changes* allows complete serial device control by the Windows application so it operates exactly like a directly connected serial COM port. It provides a complete COM port interface between the attached serial device and the network, providing hardware and software flow control. So, for example, the baud rate of the remote serial port is controlled by the settings for that COM port on Windows computer. If *Propagate local port changes* is not selected, then the serial port configuration parameters are set on the *console server*

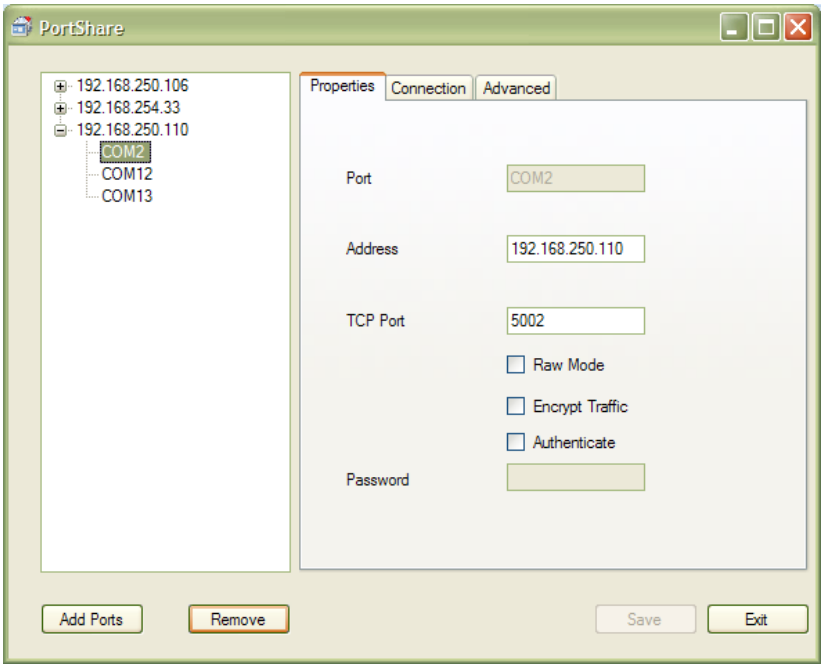
- With *Emulate Baud Rate* selected, *PortShare* will only send data out at the baud rate configured by the local application using the COM port



1.3 To remove a configured port

At any stage, you can delete a single configured COM port, or delete the *console server* connection and all the COM ports configured on that *console server*

- Select the *console server* or COM port from the left-hand menu
- Click the *Remove* button



1.4 Configure the remote serial device connection

Ensure the remote serial device is connected to your remote *console server*. Then configure the serial port as detailed in the Opengear User Manual:

- Set the serial port's Common Settings (e.g. baud rate)
- Check the *Console Server* Mode button and specify the appropriate protocol to be used:
  - *RAW TCP* allows connections directly to a TCP socket and the default TCP port address is 4000 + the serial port number (*i.e.* the address of the second serial port is *IP Address _ 4002*)
  - *RFC2217* enables serial port redirection on that port. The default TCP port address is 5000 + the serial port number(*i.e.*the address of a 48 port *console server* would be IP Address _ Port(5001 – 5048))
  - *PortShare Secure* mode enables encrypted communication

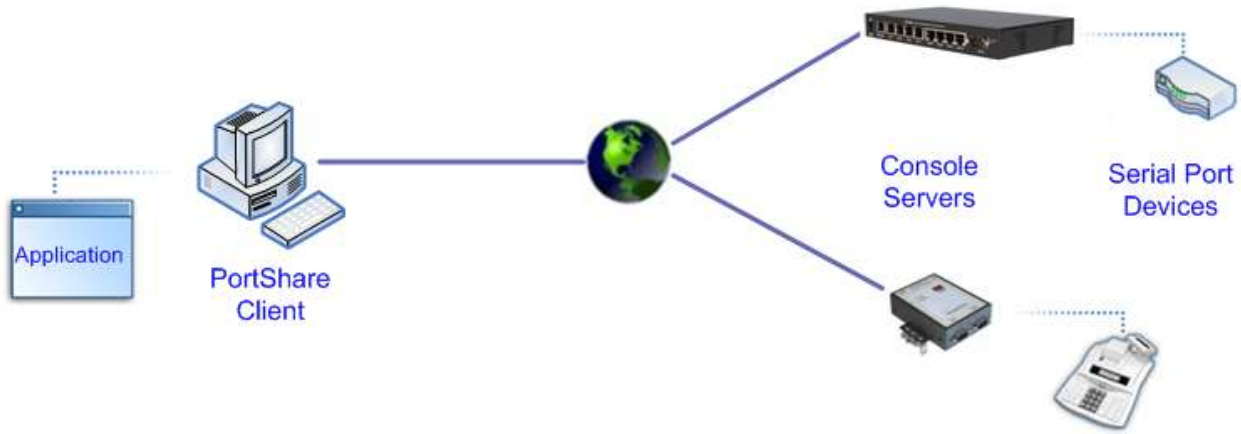
2. *PortShare* for Linux

The *PortShare* driver for Linux maps the *console server* serial port to a host *tty* port.

Opengear has released the ***portshare-serial-client*** as an open source utility for Linux, AIX, HPUX, SCO, Solaris and UnixWare. The General Public License agreement is included in Chapter 3 and the utility can be freely downloaded from ftp://ftp.opengear.com.

This *PortShare* serial port redirector allows you to use a serial device connected to the remote *console server* as if it were connected to your local serial port. The *portshare-serial-client* creates a pseudo tty port, connects the serial application to the pseudo tty port, receives data from the

pseudo tty port, transmits it to the *console server* through network and receives data from the *console server* through the network and transmits it to the pseudo-tty port.



So using this driver, you can use a remote console serial port as a local *tty* port and control remote serial devices as though they were attached locally to the Linux host. The driver can run under Linux kernel 2.4.x (supporting IPv4 only) and Linux kernel 2.6.x (supporting IPv4 and IPv6)

To map a *Console Server* serial port to a host tty port, you first need to setup the *console server* and attach and configure the serial port device:

- Ensure the *console server* IP configuration is valid and that you can access the unit successfully (eg: using ping or telnet)
- Configure the *Console Server* serial port to RAW TCP or RFC 2217 mode

Then you will need to install driver files into the host as detailed in 2.1 below.

2.1 Install the *PortShare* Linux driver

These simple installation instructions point to the appropriate configuration files and man pages:

2.1.1 To build and install the *PortShare* package (as root):

$ ./configure && make && make install

Note that the `--prefix=' option is ignored by configure.

2.1.2 Configure the devices by editing */etc/portshare-devices*. There are sample configurations in there, and the format is documented at the top of the file, or in the *portshare-devices* man page.

2.1.3 Start the *Portshare* devices:

/usr/local/sbin/portshare-serial-client start

(man portshare-serial-client for more information)

2.1.4 Useful commands:

portshare-stty

Used like 'stty', but applies the settings to the remote serial port correctly. A normal stty on /dev/ottyXX will not set the parameters on the port correctly, since it is just a pseudo-tty

2.1.4 Caveats

The local unix tty devices setup are just symlinks to pseudo-tty devices, so settings on those devices do not get set on the *Console Server.* To do this, use *'portshare-stty'*.

This also means that applications that rely on setting tty parameters such as baud rate, modem signals, etc will not work unless they are started with the *libportshare-ser-cli.so* library preloaded.

e.g. *LD_PRELOAD=/usr/local/lib/libportshare-ser-cli.so stty -a < /dev/otty01*

Use */usr/local/sbin/portshare-stty* as a template for running your application with the library preloaded.

2.2 *PortShare* command man pages

portshare-devices.txt is the man page for the portshare-devices configuration file. It is the more formal explanation without examples.

portshare-ser-cli is the man page for the binary .c program, or backend which does the actual work.

portshare-serial-client is the man page for a script. This script acts as the front end, or interface into the portshare-ser-cli binary.

2.2.1 portshare-serial-client(8)

NAME

portshare-serial-client Serial Port Interface for *Console Servers*

SYNOPSIS

portshare-serial-client (start | stop | restart | status) [devname]

DESCRIPTION

For each physical port listed inthe portshare-devices file, portshare-serial-client controls the status of the corresponding portshare-ser-cli interface

OPTIONS

portshare-serial-client must be invoked with one of the mandatory options including start, stop, restart or status and optionally with an argument associated with a specific device. In this case, the action specified as the option will be performed only for this device.

If this argument is not supplied, the action will be performed for all devices listed in portshare-devices file. The mandatory options include:

- *start* - Starts the portshare-ser-cli program, using parameters supplied in the portshare-devices file. If this program is already running, a message will be displayed, and no additional copy will be started.
- *stop* - Stops the portshare-ser-cli program(s), by issuing a SIGTERM signal.
- *restart* - Simulates a hang-up to portshare-ser-cli program(s), by issuing a SIGUSR1 signal.
- *status* - Checks the status of portshare-ser-cli programs(s).

EXAMPLE

Assuming the following portshare-devices file configuration:

/dev/ctty01:pr3k:1:rtelnet: /dev/ctty02:pr3k:2:socket:

1. Start all devices: portshare-serial-client start
   Messages: "Starting /dev/ctty01 <==> pr3k:1 interface" "Starting /dev/ctty02 <==> pr3k:2 interface"

2. Try to start them again: portshare-serial-client start
   Messages: "portshare-serial-client : /dev/ctty01 already active" "portshare-serial-client : /dev/ctty02 already active"

3. Stop only /dev/ctty01 device: portshare-serial-client stop/dev/ctty01
   Messages: "Stopping /dev/ctty01 <==> pr3k:2 interface"

4. Checking status: portshare-serial-client status
   Messages: "/dev/ctty01 (rtelnet at pr3k:1) is inactive" "/dev/ctty02 (socket at pr3k:2) active, pid 2983"

5. Start a non-valid device portshare-serial-client start /dev/xtty01
   Messages: "portshare-serial-client : device /dev/xtty01 does not exist"

2.2.2 portshare-ser-cli(8)

NAME

portshare-ser-cli Serial Port Interface for *console servers*

SYNOPSIS

portshare-ser-cli [options] devname rasname physport

DESCRIPTION

The portshare-ser-cli program connects a Unix device file, devname, to a physical port, physport, of an Opengear *console server,* rasname. portshare-ser-cli provides the I/O interface between the device file and the physical port, running as an user-mode device driver.

If physport is assigned to 0, then rasname is used as the IP address on an IP-based serial port addressing.

OPTIONS

portshare-ser-cli may be started with the following options:

- u ptyiosize
  Sets the internal device I/O size to ptyiosize (maximum: 4096 bytes, default: 1024 bytes)
- n netiosize
  Sets the internal socket I/O size to netiosize (maximum: 512 bytes, default: 128 bytes)
- i retrydelay
  Delay in seconds between connection requests (default: 60)
- r retries
  Number of connection request retries before exiting. (default: infinity)
- s
  Use the Socket Server protocol for talking to the server, this means just piping all the data down a TCP connection with no control information, so it's impossible to change the port speed etc. The default is to use the RFC2217 protocol.
- m modem handling
  The default is 0 which means to check DCD state, 1 means to ignore DCD.
- c close mode
  Last close handling; the default is 0 which means to hangup the modem, 1 means not to hangup.
- p start port
  TCP base port of servers at *console server* side (default: 5000 for RFC2217 Server). Note: if physport is assigned to zero, this option has no effect, the Telnet Server standard port (23) is used.
- d debug level
  The default is debug level 0 (little debugging), level 1 debugs internal state changes, level 2 debugs events, and level 3 debugs IO calls.
- f
  Run in foreground, this is suitable for running from init.
- x
  Console mode: normally all messages are sent to syslogd (using local2 facility). With this option, all messages will be sent to stdout and portshare-ser-cli runs in the foreground. This implies -f
- P
  Specify a TCP port to connect to. If this option is present, it will override most other options in the /etc/portshare-devices file. portshare-ser-cli will use this TCP port instead of deriving it from –p and physport. This option is useful when connecting to a local TCP port, which is connected to an ssh tunnel.

USE

Every instance of portshare-ser-cli will have a virtual serial device which is a sym-link to a pseudo-tty. A terminal program can then talk to the virtual serial device and it's data transfers

will be redirected across the network. Each virtual serial device will be accompanied by a UNIX domain socket with the same name with the addition of ".control". So if portshare-ser-cli provides the virtual device named "/dev/modem" then it will have a control socket named "/dev/modem.control". There is a shared object named libportshare-ser-cli.so which intercepts calls to the tcsetattr() and tcsendbreak(). This shared object then sends the relevant data to the portshare-ser-cli server via the control socket. To recognize a virtual modem device, it has to read /etc/portshare-devices.

The libportshare-ser-cli.so shared object can be loaded per-application through the LD_PRELOAD environment variable, or for the entire system through the system shared object configuration (see the OS documentation). Note that the LD_PRELOAD environment variable has to have the fully qualified path of the object; otherwise an application which changes its current directory may fail.

BUGS

In Solaris, libportshare-ser-cli.so does not work with the stty program. stty uses a different interface to this and requires some extra coding.

In Solaris, libportshare-ser-cli.so conflicts with some system programs, such as ps, for unknown reasons. It is advised not to load it for those programs. It has no such problems with any serial communications programs.

Example.

Start an interface between /dev/prt1 device and serial port number 10 on a *console server* named pr01, without hang-up at last close:

portshare-ser-cli -c 1 /dev/prt1 pr01 10

In general use, do not start portshare-ser-cli from the command line. Start it through the portshare-serial-client script or from init.

2.2.3 portshare-devices(5)

NAME

portshare-devices - tables for driving portshare-serial-client

DESCRIPTION

The portshare-devices file supplies all mapping between Unix device files (/dev/*) and the addresses of serial ports of *console servers*. It contains one entry for each serial port, with the following format:

device:rastype:rasname:physport:type:options

Note: A # character at beginning of line indicates a comment

The entry fields are:

devname

- The full pathname of the file that will be associated with the serial port. It must start with a "/dev/" prefix. Two naming schemes may be used here:
  - devname does not exist, and will be linked to a free pseudo-tty. This is the default behavior of portshare-ser-cli.
  - devname is the name of a valid slave pseudo-tty. In this case, the “t 1” option must be assigned in the options field. (Note: this option is not supported by this release).

rastype

- Console server type:
  - cm4001, for Opengear CM4001
  - cm4008, for Opengear CM4008
  - cm4116, for Opengear CM4116
  - cm4148, for Opengear CM4148
  - sd4002, for Opengear SD4002
  - sd4008, for Opengear SD4008
  - im4208, for Opengear IM4208
  - im4216, for Opengear IM4216
  - im4248, for Opengear IM4248
  - img4004, for Opengear IMG4004
  - acm5002, for Opengear ACM5002
  - acm5003, for Opengear ACM5003
  - acm5004, for Opengear ACM5004

rasname

- Host Name or IP Address of the *console server* where the serial port resides.

physport

- Number of physical port in the *console server*. If treated as the IP address associated with this port, in a IP-based addressing scheme.

type

- Server type that will be contacted to handle the serial port:
  - rfc2217, for RFC2217 serial support
  - rtelnet, for Remote Telnet Server - socket, for Raw TCP Socket Server

options

- Per-port specific options, passed to portshare-ser-cli program.

FILES

/etc/portshare-devices

2.3 Some *PortShare* application examples

These examples show the actual virtual port configuration and explain how the configuration should be done, using examples:

/etc/portshare-devices

Entry Syntax:

devname:cmtype:cmname:physport:type:options

where :
devname -> Device full pathname

rastype -> *Console server* type (cm4008, CM4116 or CM4148)

rasname -> *Console server* host name or IP address

physport -> Physical port number on *console server*

type -> Server type : rfc2217 or socket (raw TCP)

opts -> per-port interface options (optional)

To connect via a secure ssh tunnel, use the '-P' parameter as part of "opts", and give the TCP port number used for the local end of the tunnel. e.g. "-P 22222" will attempt to connect to local TC port 22222. Also set the rasname to "localhost". (Note: The ssh tunnel must already be setup for this to work).

Example 1.

Connect to port 1 on a 48 port *console server,* such as cm4148, at IP address 10.111.254.1, using RFC2217:

/dev/otty01:cm4148:10.111.254.1:1:rfc2217

Example 2.

Connect to port 8 on a CM4008 *console server* at IP address 10.111.254.2, using RFC2217:

/dev/otty02:cm4008:10.111.254.2:8:rfc2217

Example 3.

Create an ssh tunnel from localhost to *console server.* This tunnel connects to serial port 3 on the *console server* and uses rfc2217. Assume the rfc2217 TCP port base on the *console server* is set to the alternate value of 9000 (whereas, by default, it would be 5000). The local TCP port used for the tunnel is 12345:

ssh -L 12345:10.111.254.3:9003 <username>@10.111.254.3 -N

Now use this tunnel to make the connection:

/dev/otty03:cm4008:localhost:3:rfc2217:-P 12345

3. *PortShare* License Agreements

3.1 End User License Agreement - *PortShare* for Windows

NOTICE TO USER:
PLEASE READ THIS CAREFULLY. BY USING ALL OR ANY PORTION OF THE SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT USE THIS SOFTWARE.

1. DEFINITIONS
When used in this Agreement, the following terms shall have the respective meanings indicated, such meanings to be applicable to both the singular and plural forms of the terms defined:
"Licensor" means the company who supplied the PortShare software for use with a serial console server product manufactured by the company
"Licensee" means You or Your Company, unless otherwise indicated.
"Software" means (a) all of the contents of the files, disk(s), CD-ROM(s) or other media with which this Agreement is provided, including but not limited to ((i) registration information, i.e. License key which is unique for a registration name of the Licensee; (ii) related explanatory written materials or files ("Documentation"); and (iii) Software setup files and code samples (if any); and (b) upgrades, modified versions, updates, additions, and copies of the Software, if any, licensed to you by the Licensor (collectively, "Updates").
"Use" or "Using" means to access, install, download, copy or otherwise benefit from using the functionality of the Software in accordance with the documentation in the User Guide.
"Computer" means an electronic device that accepts information in digital or similar form and manipulates it for a specific result based on a sequence of instructions.

2. GENERAL USE
2.1 As long as the Licensee complies with the terms of this End User License Agreement (the "Agreement"), the Licensor grants the Licensee a non-exclusive right to Use the Software on their Computer(s) for the purposes of initiating local real or virtual serial port data redirection from their Computer(s) to the physical serial ports on a remote console server manufactured by the Licensor using TCP/IP protocol.
2.2 The License is royalty-free and the Software may be installed and used by the Licensee for commercial purposes or in business environment
2.3 An unlimited number of copies of the Software may be installed and used on any number of Computers at all and any offices of the Licensee's Company worldwide, provided all copies are used solely for the purposes covered in 2.1.
2.4 The Software may be installed and used by any number of users at all and any offices of the Licensee's Company worldwide provided the Software is used solely for the purposes covered in 2.1.

3. INTELLECTUAL PROPERTY RIGHTS
3.1 This License does not transmit any intellectual rights on the Software. The Software and any copies that the Licensee is authorized by the Licensor to make are the intellectual property of and are owned by the Licensor and its suppliers.
3.2 The Software is protected by copyright, including without limitation by Copyright Law and international treaty provisions.
3.3 Any copies that the Licensee is permitted to make pursuant to this Agreement must contain the same copyright and other proprietary notices that appear on or in the Software.

3.4 Trademarks shall be used in accordance with accepted trademark practice, including identification of trademarks owners' names. Trademarks can only be used to identify printed output produced by the Software and such use of any trademark does not give the Licensee any rights of ownership in that trademark.

7. WARRANTY

7.1 The Licensor warrants that:

7.1.1 it owns the Software and documentation and/or is in possession of valid and existing licenses that support the terms of this agreement,

7.1.2 to the best of its knowledge, the Software does not infringe upon or violate any intellectual property right of a third party and the Licensor will indemnify, defend and hold the Licensee harmless in the event of a claim related thereto;

7.1.3 the Software does not contain any back door, time bomb, drop dead device or other routine intentionally designed by the Licensor to disable a computer program or computer instructions that alter, destroy or inhibit the processing environment.

7.2 Except those warranties specified in section 7.1 above, the Software is being delivered to the Licensee "AS IS" and the Licensor makes no warranty as to its use or performance.

The Licensor and its suppliers do not and cannot warrant the performance or results the Licensee may obtain by using the Software.

Except for any warranty, condition, representation or term to the extent to which the same cannot or may not be excluded or limited by law applicable to the Licensee in his jurisdiction, the Licensor and its suppliers make no warranties conditions, representations, or terms (express or implied whether by statute, common law, custom, usage or otherwise) as to any matter including without limitation non-infringement of third party rights, merchantability, integration, satisfactory quality, or fitness for any particular purpose.

8. LIMITATION OF LIABILITY

In no event will the Licensor or its suppliers be liable for any damages, claims or costs whatsoever or any consequential, indirect, incidental damages, or any lost profits or lost savings, even if the Licensor has been advised of the possibility of such loss, damages, claims or costs or for any claim by any third party. In no event will the Licensee be liable to The Licensor on condition that the Licensee complies with all terms and conditions stated in this License.

9. NON-WAIVER

If a portion of this agreement is held unenforceable, the remainder shall be valid. It means that if one section of the Agreement is not lawful, the rest of the Agreement is still in force. A party's failure to exercise any right under this Agreement will not constitute a waiver of (a) any other terms or conditions of this Agreement, or (b) a right at any time thereafter to require exact and strict compliance with the terms of this Agreement.

3.2 End User License Agreement - *PortShare* for Linux

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER

EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS